

OPMとScorecarding法の組合せによるシステム開発手法の評価

Evaluation of combining OPM and Scorecarding in System Development

岡 啓[†] 当麻 哲哉[†]
Akira Oka[†] Tetsuya Toma[†]

[†]慶應義塾大学大学院 システムデザイン・マネジメント研究科

[†] Graduate School of System Design and Management, Keio University

要旨

情報システム開発において要求の分析や改善に関する手法は様々提案されている。しかし、いくら正確な要求仕様を作成しても、現実には仕様変更を避けることは難しく、むしろ仕様変更による手戻りを抑制することの方が重要である。本稿では、シンプルなモデリング言語 OPM(Object-Process Methodology)とゴール指向的な Scorecarding 手法の組合せにより、仕様変更に起因する手戻りを抑制し、開発の効率性が向上することを示すとともに、こうした手法の評価に、見積手法の1つである COSMIC 法が有効であることを示すものである。

1. はじめに

今日のビジネス環境の変化を考慮すれば、情報システムにおける要求の変更は不可避であり、どれほど正確に要求から仕様を作成されていたとしても、仕様変更を避けることは難しい。そうであるならば、仕様変更による手戻りを少なくする設計手法が現実的に求められているのではないだろうか。実際、熟練のエンジニアは効率的な設計、即ち、手戻りの少なくなる設計を行っている。変更に対するロバスト性をもつ設計ができれば、手戻りを少なく出来るのではないかと、というのが本稿の出発点である。

では、どうすれば変更を考慮した設計になるのであろうか。逆に考えると、変更を考慮した設計ができないのは、変更可能性のある箇所を知らないからである。システムには、それによって達成させるべき目的がある。その目的の範囲内で要求が設定され、仕様が定められていく。従って、目的とその環境を知っていれば、変更可能性を考慮することが可能である。しかしながら、システム開発の現場では上流工程・下流工程といった水平分業が進んでおり、下流工程には、目的が伝わりにくい。そもそも情報を整理し精練して下流工程に伝えるのが上流工程の役目でもある。従って、これまでとは異なる、上流工程から下流工程へ目的を伝達する方法を考える必要がある。

目的を明らかにするという点では、ゴール指向と呼ばれる手法類が注目されている[1]。但し、ゴール指向は、ゴールや目的が顧客要求の文脈で語られてきたため、システム構成要素や設計と言った視点まであまり想定されていない。他方、モデリング言語は、上流工程から下流工程まで階層的に対応しているが、目的や冗長な情報を持ち合わせていない。そこで、異なる利点も持つモデリング言語とゴール指向手法の組合せにより下流工程でも目的がトレース可能な手法の構築を本稿では試みる。

上記の目的のために、OPM(Object-Process Methodology)と Scorecarding 法の組合せを選択した。理由は、これらの手法がシンプルであり、導入や検証がし易いからである。各手法の概要については後述する。

2. 先行事例

目的を明確化するという点に着目した仕様表現手法として、清水吉男氏の提唱する USDM 表記法 (Universal Specification Description Manner)がある [2]。USDM 法は、(財)日本情報システム・ユーザー協会でも推奨され[3]、成果に関する報告もある[4]。USDM 法では、仕様を記述する際に理由も併記する手法である。また、清水氏も「仕様の変化の方向に対する予測」の重要性を指摘している[2,p103]。こうした点は、本稿で試みる手法と類似する。一方、USDM 法が仕様書という文書がベースなのに対し、本稿では、モデリング言語を通じたエンジニアリングの確立を目指している。また、派生開発用の手法である XDDP(eXtreme Derivative Development Process)[5]も提唱されているが、本稿で扱う変更可能性とは異なる。本稿で扱う変更可能性とは、新規開発や派生開発に区別されない、設計上の課題である。

3. 手法の個別説明

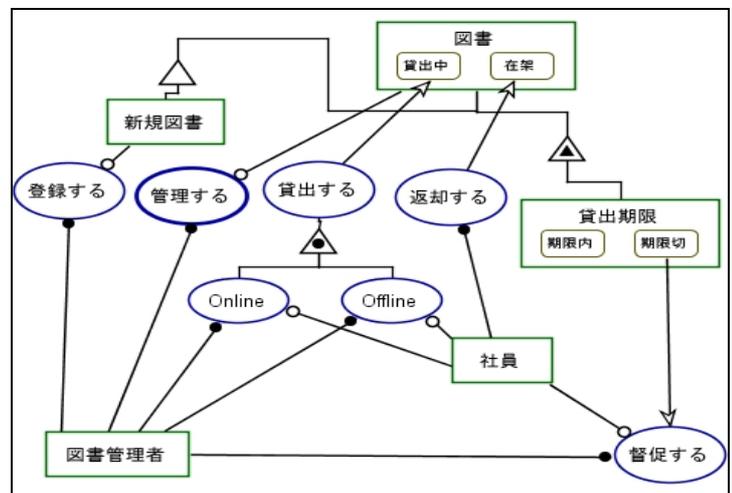
3.1. OPM(Object-Process Methodology)

OPMとは、イスラエル工科大学のDov Dori教授によって考案されたシステム・モデリング手法であり [6]、オブジェクトとプロセスの関係を決定する事によりシステム全体を表現できる。同様にシステム全体をモデリングできる手法としてSysMLがある。SysMLはUMLを拡張する形で体系化されているため、IT技術者には習得しやすいが、逆にUMLの知識がない場合は、ダイアグラムの種類が多いため、習得に時間がかかる。一方、OPMは、UMLとは異なる表記方法を採用しており、単一のダイアグラムであるため、UMLの知識がない人でも容易に習得できる[7]。また、OPMではダイアグラム(OPD)とランゲージ(OPL)の両方で同一内容を記述でき、OPDを初めて見た場合でもOPLより容易に意味を理解できる。更に、オブジェクトとシステムを階層的に記述できるので、システム全体からサブモジュールまでモデリング可能である。一般的なOPMによる表記事例を図1と表1に示す。詳細な説明は省くが、四角がオブジェクトであり、丸がプロセスである。

表1 OPLの例

図書 can be 貸出中 or 存架.
図書 exhibits 貸出期限
貸出期限 can be 期限内 or 期限切.
社員 handles 返却する.
図書管理者 handles 登録する, 管理する, and Offline
新規図書 is a 図書.
貸出する yields 貸出中 図書.
登録する requires 新規図書.
返却する yields 在架 図書.
督促する requires 社員.
督促する consumes 期限切 貸出期限.
Online is instance of 貸出する.
※一部省略した

図1 OPDの例



3.2. Scorecarding 法

Scorecarding法とは、文字通りスコア（点数）を記録する手法である。経営工学におけるバランスト・スコア・カードや生産工学におけるシックスシグマなどに組み込まれている手法が有名である。ここでは、Stanford大学で研究されているQuality Scorecarding法やProject Scorecarding法をベースとして[8]、それをシステムとその構成要素に適用することを試みる。本来は、システム全体を評価するためのものであるが、システムの構成要素はサブシステムとして階層的に構成されるのが常であるので、同様にScorecarding法も階層的に適用可能である。具体的には、表2の評価項目とその内容を洗い出し、構成要素をサブシステムと見なして、同様の評価を階層的に適用する。

表2 Scorecarding 法の評価項目

記号	評価項目	説明
Y	目的・理由(Objective Measures)	システム、サブシステム、機能の目的
X	制御可能要素(Control Factors)	Yにとって制御可能な要素
V	制御不可能要素(Noise/Uncertainty Factor)	Yにとって制御不可能な外乱・不確定要素
F	変換関数(Transfer function)	Y=F(X,V)の関係が成立する評価項目間関係

なお、今回は評価項目の分類のみを行い、評価項目の数値化（重みなど）や階層を跨ぐような相互関係は分析しない。

3.3. COSMIC 法

COSMIC 法 (Common Software Measurement International Consortium 法) とは、従来 COSMIC - FFP 法として知られていた機能規模測定法である[9]。ISO/IEC19761 として国際規格化されており、ここでは詳細の説明を省く。簡単に説明すると、外部環境とのエントリーとイグジット、内部環境へのリードとライトの数をデータの種類ごとにカウントする手法である。

今回 COSMIC 法を使用することにしたのは、効果の確認を定量的に測定する必要性があり、COSMIC 法では、比較的簡単に実施できることと、OPM の構造から計算がしやすいと考えたからである。

4. 組合せによる手法

通常モデリングの他に、スコアの評価項目を OPM で記述する。評価項目と OPM での表記方法は、表3の通りである。通常モデリングと区別するために、接頭辞として記号をつけることにした。簡単な事例を図3と表4に示す。

表3 Scorecarding 法と OPM の対応

記号	評価項目	OPM 上の表記法
Y	目的・理由(Objective Measures)	プロセスとして表現
X	制御可能要素(Control Factors)	オブジェクトとして表現
V	制御不可能要素(Noise/Uncertainty Factor)	オブジェクトとして表現
F	変換関数(Transfer function)	オブジェクトとプロセスの関係により線によって表現

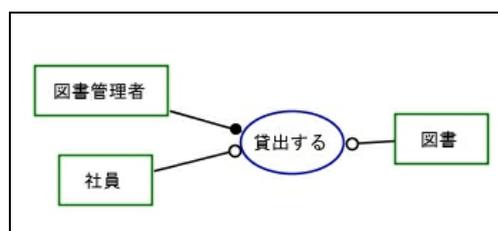


図2 通常の OPM によるモデリング

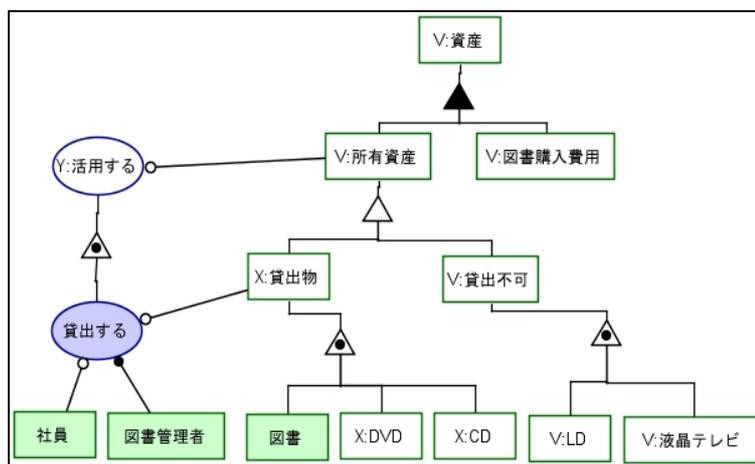


図3 Scorecarding 付 OPM

表4 図3の OPL による表現

図書管理者 handles 貸出する.
V:資産 consists of V:図書購入費用 and V:所有資産.
X:CD is instance of a X:貸出物.
X:DVD is instance of a X:貸出物.
図書 is instance of a X:貸出物.
V:LD is instance of a V:貸出不可.
X:貸出物 is a V:所有資産.
V:貸出不可 is a V:所有資産.
V:液晶テレビ is instance of a V:貸出不可.
Y:活用する requires V:所有資産.

貸出する is instance of Y:活用する.

貸出する requires X:貸出物 and 社員.

この例では、CDやDVDが貸し出し対象になり得るがLDは対象外であることが表現されている

5. 検証

情報システムの事例として、簡単な図書管理システムを用いる。JUAS・UVC 報告書の図書管理システムの事例[3,p121]を参考に設定した。既にOPMの事例として本稿に登場しているが、次のような機能を有している。

- 1) 新規書籍の登録
- 2) 書籍の管理
- 3) オンライン貸し出し処理
- 4) 資料室貸し出し処理
- 5) 返却処理
- 6) 督促処理

上記の初期機能に対し、下記のような仕様変更を設定した。

- 7) CD・DVDの貸し出し追加
- 8) 社員以外への貸し出し

以上の仕様変更があった場合に、初期の設計の違いが後の修正規模にどのように影響するか、Scorecarding付OPMと通常のOPMをCOSMIC法にて比較した。なお、COSMIC法で評価する対象は、実際の成果物であるから、Scorecarding付OPMであってもそこから参考情報であるScorecardingを除いたものが対象となる。

今回の検証により、モデリング言語とゴール指向手法の組合せによる有効性が基本的な部分で確認できた。即ち、目的を下流工程まで伝える仕組みを構築することにより、下流工程での設計に差異が見られた。

6. まとめ

これまでの取り組みにより、本稿の目標とした変更可能性を考慮した設計の改善と開発手法に関する定量的な評価の礎を築くことができた。

残された課題として、事前の準備や教育のコストを測定する必要がある。また、どのようなケースで効果的なのか事例を重ねて検証するとともに、より高機能なモデリング言語やゴール指向手法との連携も検討していくべきである。

参考文献

- [1] 山本修一郎, ゴール指向による!! システム要求管理技法, ソフト・リサーチ・センター, 2007.
- [2] 清水吉男,[入門+実践]要求を仕様化する技術 表現する技術、技術評論社、2005
- [3] (社)日本情報システム・ユーザー協会, 要求仕様ガイドライン JUAS・UVC 研究プロジェクト報告書, (社)日本情報システム・ユーザー協会,2007
- [4] 中井栄次,間瀬研二,古畑慶次,“無知見プロジェクトに対する XDDP の適用- USDm、プロセス定義によるプロセス改善-“,SQiP シンポジウム 2009,2009
- [5] 清水吉男,「派生開発」を成功させるプロセス改善の技術と極意、技術評論社、2007
- [6] Dov Dori, Object-Process Methodology :A Holistic Systems Paradigm, Springer, 2002.
- [7] Grobstein, Y. Perelman, V. Safra, E. Dori, D., "Systems Modeling Languages: OPM Versus SysML", Systems Engineering and Modeling, 2007. ICSEM '07. International Conference, 2007
- [8] Lawrence P. Chao, Kosuke Ishii, "DESIGN PROCESS ERROR-PROOFING: CHALLENGES AND METHODS IN QUANTIFYING DESIGN ELEMENTS", Tenth ISSAT International Conference on Reliability and Quality in Design, 2004
- [9] The Common Software Measurement International Consortium, The COSMIC Functional Size Measurement Method Version 3.0.1 Measurement Manual (The COSMIC Implementation Guide for ISO/IEC 19761: 2003), May 2009